CpSc 2120 — Goddard — Spring'18

Lab 2: Arrays/Vectors and Classes

Create a class PosPoly that stores polynomials in x with positive integer coefficients. For example, $4x^5 + 2x^6 + 3x^2 + x$.

The instance variables of your class should be (a) a counter for the number of powers; and (b) an array/vector of Pair's, where a Pair is a struct (or a class) that stores two ints: the power and the coefficient. For example, the above polynomial could be stored as:

powerCount = 4

power	5	6	2	1	⋯
coeff	4	2	3	1	⋯

You can assume that no polynomial will ever have more than 100 terms.

The class should be stored in files PosPoly.cpp and PosPoly.h. (The code for Pair can be in its own files, but it is also okay to include it in PosPoly.h. It is also okay to adapt/use the Pair class on the class website as solution to Practice 2.)

Your PosPoly class should have the following methods:

- A default constructor that initializes the polynomial to zero
- An overloaded << operator for output. (It's okay if it prints out in any order and doesn't have all the bells and whistles; e.g. prints above polynomial with `+1x^1`)
- void incrementBy(int c, int p): increment the current polynomial by cx^p, where c should be positive and p should be nonnegative.
  For example,

  ```
  PosPoly A;
  A.incrementBy(3,2);
  A.incrementBy(2,6);
  A.incrementBy(1,5);
  A.incrementBy(1,1);
  A.incrementBy(3,5);
  ```

  should produce the above example polynomial.
- A boolean test for whether two polynomials are equal

A sample test driver is provided. Adapt as desired. (Do not add main to PosPoly.cpp.)

Submit via handin the files PosPoly.h/cpp (and Pair.h/cpp if created).
(Your driver will not be used in grading.)